Projekt 2: Entwicklung der Geschäftslogik (Hexagonale Architektur)

Autor: Andreas Kleinbichler

Datum: 20.12.2021

Thema

Bei diesem Projekt geht es darum die Geschäftslogik (Domäne) eines Microservices zu implementieren ohne dabei seine Außenwelt (Infrastruktur) zu berücksichtigen.

Um eine Fokussierung auf die reine Geschäftslogik herzustellen, wird ein hexagonaler Ansatz gewählt. In diesem Sinne werden alle Abhängigkeiten zur Infrastruktur (MessageBus, Datenbank …) durch passende Interfaces abstrahiert. Die Implementierung der Adaptoren, also die Einbindung konkreter Infrastruktur (e.g.: MongoDB, Rabbitq, MySQL, Kafka …) ist nicht Teil des Projektes.

Die Geschäftslogik (SellTicket Service)

In unserem System (***Second Hand Ticket App*** aus dem 1. Projekt) gibt es ein Microservice, das den Verkaufsvorgang von Tickets realisiert.

Das Verkaufsszenario

Kunde A verkauft zum Preis P ein Ticket T an den Kunden B. Dabei wird dem Kunden B das Geld von seinem Konto abgebucht und dem Kunden A gutgeschrieben. Dieser Transfer muss aber mit dem Verkauf des Tickets abgestimmt werden (Stichwort Distributed Consensus).

Anforderungen an das Verkaufsszenario

- Es kann ein Ticket nur einmal verkauft werden.
- Es darf dem Kunden A nur Geld zugeschrieben werden, wenn dieses von B abgebucht wurde.
- Solange das Geld nicht an Kunde A überwiesen wurde, darf das Ticket nicht als verkauft markiert werden.
- Im Falle eines erfolgreichen Verkaufsvorganges soll ein Event für die Darstellung in Social Media Plattformen gepublished werden. Dieses soll die beiden Kunden (KundeId), das Ticket (TicketId) und den Preis enthalten.
- Unabhängig davon ob der Verkaufsvorgang abgeschlossen werden kann, sollen Informationen für Metriken gepublished werden. Konkret soll es einen Counter für erfolgreiche und abgebrochene Verkaufsfälle geben.

Folgende Services sind involviert

- ***TicketSellService*** -> agiert als Transaktionskoordinator des Ticketverkaufs (die Geschäftslogik dieses Service ist Inhalt dieses Projektes)
- ***AccoutingService*** -> Bucht den Kundenkonten Geld auf und ab (muss nicht implementiert werden sodern wird nur über ein passendes Interface abstrahiert)
- ***TicketShelfService*** -> Verwaltet den Verkaufsstatus des Tickets (muss nicht implementiert werden, sondern wird nur über ein passendes Interface abstrahiert)

Zur Aufgabe

- Implementierung der Geschäftslogik des ***SellTicketServices***.

- Für die benötigten Abhängigkeiten (e.g.: AccountingService, TicketShelfService, EventPublishing) müssen passende Interfaces entworfen werden. Die Implementierung derselben, da diese ja außerhalb als Adaptoren zu Verfügung gestellt werden, ist nicht Teil des Projektes.
- Programmiersprache ist frei wählbar. Falls man aber etwas Anderes als Java, C#, C++, Go oder node.js verwenden möchte, bitte nochmals kurz Rücksprache halten.